the data into FIFO 18 in packets. When a data packet is present in FIFO 18, DMA Rx channel 29 may retrieve it, may write the packet to RAM 24, and may signal CPU 22 that data was received. CPU 22 may convert the packet data to the PHY format and may send it to the network.

[0023] When a packet is received by the device from the network, CPU 22 may be instructed to retrieve the packet. CPU 22 may convert the packet according to the PHY protocol and may write the converted packet data to RAM 24. CPU 22 may also write to common register 10, information that may be needed by CPU 12, such as timing decisions. CPU 22 may instruct Tx channel 21 to send the packet to CPU 12. Tx channel 21 may retrieve the packet data from RAM 24 and may write it to FIFO 28. Rx channel 19 may read the data from FIFO 28 and may write it to RAM 14. Rx channel 19 may then send a signal to CPU 12 that a packet has been received. CPU 12 may further process the data according to the MAC protocol and may inform the device that network data has been received.

[0024] It is noted that the channels of DMA 16/26 are not limited to either receive or transmit functionality. Rather, it is for the duration of a given transaction that a channel may only communicate in one direction.

[0025] In a further embodiment of the present invention, information may be sent only in one direction between multiple CPUs, and thus only one FIFO may be required. There are graphic applications, for example, in which no data is received from the screen. In such a case, data may only be sent from CPU 12 to CPU 22. CPU 12 may instruct Tx channel 11 to retrieve the data from the correct location in RAM 14 and to write it into FIFO 18 as described hereinabove. Rx channel 29 may retrieve the data, may write it to RAM 24, and may signal CPU 22 as described hereinabove.

[0026]If no data is sent from the screen, it may not be necessary to include FIFO 28 or the DMA channels controlling it, Tx channel 21 or Rx channel 19. Thus, an embodiment comprising a single FIFO and DMAs comprising a single channel is within the scope of the present invention.

[0027]Fig. 3, to which reference is now made, is a block diagram of exemplary FIFO 18, in accordance with an embodiment of the present invention. FIFO 18 may comprise control 32, data block 34, and side information block 36, and may process various signals. For example, control 32 may receive write (WR) signals from, and may output FIFO full signals to, Tx channel 11. Control 32 may also receive read (RD) signals from, and may output FIFO empty signals to, Rx channel 21. Data block 34 may receive data from system data bus 17 and may in turn output data to system data bus 27. Side information block 36 may receive an EOP_in signal from Tx channel 11 and may output an EOP_out signal to Rx channel 21. An EOP_out signal may alert Rx channel 21 that all data has been read.

[0028]It is noted that the specific signals given in the embodiments described herein are only examples of the types of signals that may be used. Other signals may be used as appropriate and are within the scope of this invention.

[0029]Data block 34 may comprise a predetermined number of rows, for example 2-4. Each row may be a full data row 38, a last data row 40, or a delimiter row 42. Each row in data block 34 may comprise a predetermined number of bits, for example, 128 bits. Delimiter row 42 may contain the number of the last byte containing valid data in last data row 40. Delimiter row 42 may also contain other required statuses.

[0030]Side information block 36 may comprise rows of end of packet (EOP) flags 44. There may be one row in side information block 36 for each row in data block 34.

FIFO 18 may handle side information block 36 as if it were part of data block 34. The row in side information block 36 may be as long as necessary to contain the flag. For example, EOP flag 44 may be one bit long and thus restricted to a 0/1 Boolean value. In another example, the EOP flag may indicate start, middle, and end conditions, necessitating at least two bits. Further embodiments are possible and are within the scope of this invention. For clarity purposes only, a simple 0/1, EOP false/true flag is used in the description hereinbelow. Upon receipt of an EOP_in signal, a "1" may be written in the row of EOP flags 44 corresponding to delimiter row 42. In all other rows, EOP flags 44 contains a "0".

[0031] Data may be written into data block 34 as it is received, until either a FIFO full signal is received or an end of transmission is reached. Upon receipt of a FIFO full signal, data writing may halt until FIFO 18 is no longer full. When the end of transmission is reached, an EOP_in signal may be asserted and a value may be written into delimiter row 42. Last data row 40 may contain invalid data if the written data does not fill the entire row. The value of delimiter row 42 may be used to find the number of the last byte of valid data.

[0032] In a further embodiment of the present invention, data may be word aligned. In such a case, delimiter row 42 may not be necessary and the end of packet may be indicated in last data row 40.

[0033] Data may be read from data block 34 until either a FIFO empty signal is received or an EOP_out signal is received. Upon receipt of a FIFO empty signal, data reading may halt until FIFO 18 contains data. When EOP flags 44 contains the value corresponding to end of data, in our example a 1, an EOP_out signal may be transmitted. Upon receipt of an EOP_out signal, the value contained in delimiter